Terms used
task management and **programmable network** and **ring** and **processor**

Found **66,298** of **143,484**

Sort results by: relevance
Display results: expanded form

Save results to a Binder
Search Tips
☐ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next
Best 200 shown Relevance scale

1 Emerging areas: Programming challenges in network processor deployment
Chidamber Kulkarni, Matthias Gries, Christian Sauer, Kurt Keutzer
October 2003 **Proceedings of the 2003 international conference on Compilers, architectures and synthesis for embedded systems**
Full text available: pdf(234.71 KB) Additional Information: full citation, abstract, references, index terms

Programming multi-processor ASIPs, such as network processors, remains an art due to the wide variety of architectures and due to little support for exploring different implementation alternatives. We present a study that implements an IP forwarding router application on two different network processors to better understand the main challenges in programming such multi-processor ASIPs. The goal of this study is to identify the elements central to a successful deployment of such systems based on ...

Keywords: IPv4 forwarding, mapping, multi-threading, programming heterogeneous architectures, programming model, resource sharing

2 Distributed operating systems
Andrew S. Tanenbaum, Robbert Van Renesse
December 1985 **ACM Computing Surveys (CSUR)**, Volume 17 Issue 4
Full text available: pdf(5.49 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Distributed operating systems have many aspects in common with centralized ones, but they also differ in certain ways. This paper is intended as an introduction to distributed operating systems, and especially to current university research about them. After a discussion of what constitutes a distributed operating system and how it is distinguished from a computer network, various key design issues are discussed. Then several examples of current research projects are examined in some detail ...

3 mpC: a multi-paradigm programming language for massively parallel computers
Alexey L. Lastovetsky
February 1996 **ACM SIGPLAN Notices**, Volume 31 Issue 2
Full text available: pdf(866.98 KB) Additional Information: full citation, abstract, index terms

Currently, programming systems for distributed memory machines are limited to either task parallelism or data parallelism. The mpC programming language and its programming

system support both task and data parallelism, allows both static and dynamic process and communication structures, enables optimizations aimed at both communication and computation, and supports modular parallel programming and the development of a library of parallel programs. The mpC language is an ANSI C superset. It is ba ...

4 A task- and data-parallel programming language based on shared objects
Saniya Ben Hassen, Henri E. Bal, Ceriel J. H. Jacobs
November 1998 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 20 Issue 6

Full text available: pdf(434.44 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Many programming languages support either task parallelism, but few languages provide a uniform framework for writing applications that need both types of parallelism or data parallelism. We present a programming language and system that integrates task and data parallelism using shared objects. Shared objects may be stored on one processor or may be replicated. Objects may also be partitioned and distributed on several processors.Task parallelism is achieved by forking processes remotely a ...

Keywords: data parallelism, shared objects, task parallelism

5 A proposal for certain process management and intercommunication primitives
Gary D. Knott
October 1974 **ACM SIGOPS Operating Systems Review**, Volume 8 Issue 4

Full text available: pdf(2.52 MB) Additional Information: full citation, references, citings

6 A high-speed network interface for distributed-memory systems: architecture and applications
Peter Steenkiste
February 1997 **ACM Transactions on Computer Systems (TOCS)**, Volume 15 Issue 1

Full text available: pdf(993.12 KB) Additional Information: full citation, abstract, references, index terms, review

Distributed-memory systems have traditionally had great difficulty performing network I/O at rates proportional to their computational power. The problem is that the network interface has to support network I/O for a supercomputer, using computational and memory bandwidth resources similar to those of a workstation. As a result, the network interface becomes a bottleneck. In this article we present an I/O architecture that addresses these problems and supports high-speed network I/O on dist ...

Keywords: I/O architecture, application-managed I/O, data reshuffling, distributed memory systems, network interface, outboard buffering, protocol processing, resource management

7 Programming languages for distributed computing systems
Henri E. Bal, Jennifer G. Steiner, Andrew S. Tanenbaum
September 1989 **ACM Computing Surveys (CSUR)**, Volume 21 Issue 3

Full text available: pdf(6.50 MB) Additional Information: full citation, abstract, references, citings, index terms, review

When distributed systems first appeared, they were programmed in traditional sequential languages, usually with the addition of a few library procedures for sending and receiving messages. As distributed applications became more commonplace and more sophisticated, this ad hoc approach became less satisfactory. Researchers all over the world began

designing new programming languages specifically for implementing distributed applications. These languages and their history, their underlying pr ...

8 Process migration

September 2000 **ACM Computing Surveys (CSUR)**, Volume 32 Issue 3

Full text available: pdf(1.24 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Process migration is the act of transferring a process between two machines. It enables dynamic load distribution, fault resilience, eased system administration, and data access locality. Despite these goals and ongoing research efforts, migration has not achieved widespread use. With the increasing deployment of distributed systems in general, and distributed operating systems in particular, process migration is again receiving more attention in both research and product development. As hi ...

Keywords: distributed operating systems, distributed systems, load distribution, process migration

9 Static scheduling algorithms for allocating directed task graphs to multiprocessors

Yu-Kwong Kwok, Ishfaq Ahmad

December 1999 **ACM Computing Surveys (CSUR)**, Volume 31 Issue 4

Full text available: pdf(723.58 KB) Additional Information: full citation, abstract, references, citings, index terms

Static scheduling of a program represented by a directed task graph on a multiprocessor system to minimize the program completion time is a well-known problem in parallel processing. Since finding an optimal schedule is an NP-complete problem in general, researchers have resorted to devising efficient heuristics. A plethora of heuristics have been proposed based on a wide spectrum of techniques, including branch-and-bound, integer-programming, searching, graph-theory, randomization, genetic ...

Keywords: DAG, automatic parallelization, multiprocessors, parallel processing, software tools, static scheduling, task graphs

10 Evolution of Data-Base Management Systems

James P. Fry, Edgar H. Sibley

January 1976 **ACM Computing Surveys (CSUR)**, Volume 8 Issue 1

Full text available: pdf(2.63 MB) Additional Information: full citation, references, citings, index terms

11 Current research in computer networks: a personal view

Colin Whitby-Strevens

April 1976 **ACM SIGCOMM Computer Communication Review**, Volume 6 Issue 2

Full text available: pdf(2.02 MB) Additional Information: full citation, references

12 Computer Communication Networks: Approaches, Objectives, and Performance Considerations

Stephen R. Kimbleton, G. Michael Schneider

September 1975 **ACM Computing Surveys (CSUR)**, Volume 7 Issue 3

Full text available: pdf(3.99 MB) Additional Information: full citation, references, citings, index terms

13 Paradigms for process interaction in distributed programs
Gregory R. Andrews
March 1991 **ACM Computing Surveys (CSUR)**, Volume 23 Issue 1

Full text available: pdf(3.77 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Distributed computations are concurrent programs in which processes communicate by message passing. Such programs typically execute on network architectures such as networks of workstations or distributed memory parallel machines (i.e., multicomputers such as hypercubes). Several paradigms—examples or models—for process interaction in distributed computations are described. These include networks of filters, clients, and servers, heartbeat algorithms, probe/echo algorithms, broa ...

Keywords: clients and servers, distributed and parallel algorithms, distributed programming, distributed programming methods, heartbeat algorithms, networks of filters, patterns for interprocess communication, probe/echo algorithms, replicated servers, token-passing algorithms

14 Pen computing: a technology overview and a vision
André Meyer
July 1995 **ACM SIGCHI Bulletin**, Volume 27 Issue 3

Full text available: pdf(5.14 MB) Additional Information: full citation, abstract, citings, index terms

This work gives an overview of a new technology that is attracting growing interest in public as well as in the computer industry itself. The visible difference from other technologies is in the use of a pen or pencil as the primary means of interaction between a user and a machine, picking up the familiar pen and paper interface metaphor. From this follows a set of consequences that will be analyzed and put into context with other emerging technologies and visions.Starting with a short historic ...

15 Special session on reconfigurable computing: The happy marriage of architecture and application in next-generation reconfigurable systems
Ingrid Verbauwhede, Patrick Schaumont
April 2004 **Proceedings of the first conference on computing frontiers on Computing frontiers**

Full text available: pdf(398.28 KB) Additional Information: full citation, abstract, references, index terms

New applications and standards are first conceived only for functional correctness and without concerns for the target architecture. The next challenge is to map them onto an architecture. Embedding such applications in a portable, low-energy context is the art of molding it onto an energy-efficient target architecture combined with an energy efficient execution. With a reconfigurable architecture, this task becomes a two-way process where the architecture adapts to the application and vice-vers ...

Keywords: embedded, real-time systems

16 The family of concurrent logic programming languages
Ehud Shapiro
September 1989 **ACM Computing Surveys (CSUR)**, Volume 21 Issue 3

Full text available: pdf(9.62 MB) Additional Information: full citation, abstract, references, citings, index terms

Concurrent logic languages are high-level programming languages for parallel and distributed systems that offer a wide range of both known and novel concurrent programming techniques. Being logic programming languages, they preserve many advantages of the abstract logic programming model, including the logical reading of programs and computations, the convenience of representing data structures with logical terms and manipulating them using unification, and the amenability to metaprogrammin ...

17 "Topologies"—distributed objects on multicomputers

Karsten Schwan, Win Bo

May 1990 **ACM Transactions on Computer Systems (TOCS)**, Volume 8 Issue 2

Full text available: pdf(3.83 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Application programs written for large-scale multicomputers with interconnection structures known to the programmer (e.g., hypercubes or meshes) use complex communication structures for connecting the applications' parallel tasks. Such structures implement a wide variety of functions, including the exchange of data or control information relevant to the task computations and/or the communications required for task synchronization, message forwarding/filtering under program control, and so o ...

18 Parallel execution of prolog programs: a survey

Gopal Gupta, Enrico Pontelli, Khayri A.M. Ali, Mats Carlsson, Manuel V. Hermenegildo

July 2001 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 23 Issue 4

Full text available: pdf(1.95 MB) Additional Information: full citation, abstract, references, citings, index terms

Since the early days of logic programming, researchers in the field realized the potential for exploitation of parallelism present in the execution of logic programs. Their high-level nature, the presence of nondeterminism, and their referential transparency, among other characteristics, make logic programs interesting candidates for obtaining speedups through parallel execution. At the same time, the fact that the typical applications of logic programming frequently involve irregular computatio ...

Keywords: Automatic parallelization, constraint programming, logic programming, parallelism, prolog

19 Self-stabilizing symmetry breaking in constant-space (extended abstract)

Alain Mayer, Yoram Ofek, Rafail Ostrovsky, Moti Yung

July 1992 **Proceedings of the twenty-fourth annual ACM symposium on Theory of computing**

Full text available: pdf(1.56 MB) Additional Information: full citation, abstract, references, citings, index terms

We investigate the problem of self-stabilizing round-robin token management scheme on an anonymous bidirectional ring of identical processors, where each processor is an asynchronous probabilistic (coin-flipping) finite state machine which sends and receives messages. We show that the solution to this problem is equivalent to symmetry breaking (i.e., leader election). Requiring only constant-size messages and message-passing model has practical implications: our solution can be implemented ...

20 File servers for network-based distributed systems

Liba Svobodova

December 1984 **ACM Computing Surveys (CSUR)**, Volume 16 Issue 4

Full text available: pdf(4.23 MB) Additional Information: full citation, references, citings, index terms, review

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Refine Search

Search Results -

Term	Documents
(19 AND 12).USPT.	4
(L12 AND L19).USPT.	4

Database:
US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search:

Refine Search

Recall Text Clear Interrupt

Search History

DATE: Friday, October 15, 2004 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT; PLUR=YES; OP=ADJ			
L20	l12 and L19	4	L20
L19	l10 and L18	6	L19
L18	l3 and L17	19	L18
L17	l2 and L16	92	L17
L16	shared resource$1	3366	L16
L15	l8 and L14	0	L15
L14	critical near3 instruction$1	515	L14
L13	l11 and L12	0	L13
L12	suspend$ and resum$	16226	L12
L11	l8 and L10	26	L11
L10	network$1	301135	L10
L9	l7 and L8	0	L9
L8	l5 and l6	32	L8

L7	critical instruction$1	131	L7
L6	instruction$1	218383	L6
L5	l2 and l3 and L4	33	L5
L4	(ring$ or loop$) near3 process$	33928	L4
L3	multi process$	6827	L3
L2	dependen$ near5 task$1	1112	L2
L1	6182210.pn.	1	L1

END OF SEARCH HISTORY